

Sélecteurs CSS

CSS



In CSS, selectors are patterns used to select the element(s) you want to style

Quelques liens :

[Liste des sélecteurs](#)

[Sélecteurs CSS](#)

Rappel

Une déclaration CSS comporte un sélecteur suivi d'une liste de couples « propriété valeur »:

```
sélecteur { propriété2 : valeur2 ; ... }
```

En CSS, un sélecteur est donc utilisé pour cibler une(des) partie(s) spécifique(s) d'une page web à mettre en forme. Afin de pouvoir être précis, CSS est très riche en sélecteurs, des premiers sélecteurs du CSS1 aux pseudos éléments de CSS3, en attendant les dernières nouveautés promises par CSS4.

Lors du TP précédant, nous avons déjà croisé les sélecteurs directs, d'id, de classe et de descendances.

Question 0

Pour les exercices qui vont suivre, nous vous proposons d'utiliser le code html de la page d'accueil du département Informatique. <https://www.iut.u-bordeaux.fr/info/>

Pour cela, récupérez le source html, supprimez toutes feuilles de styles, insérez une référence à votre propre feuille de style que vous complétez au fur et à mesure des questions.

Les premiers sélecteurs

- Ecrire en rouge le contenu de la (des) balise(s) h1, en fuchsia le contenu de la (des) balise(s) h2.
- Colorer en orange le contenu de la balise d'identifiant entreprises.
- Colorer en olive les éléments de class pfblock-subtitle.
- En faisant une recherche sur le selecteur universel, appliquez une marge intérieure et extérieure de 0 à tous les éléments. Pourquoi faire cela ? Est-ce une bonne pratique ?

Pour appliquer le même style à plusieurs éléments, il est possible de combiner plusieurs sélecteurs en les séparant par une virgule

- En utilisant la propriété `display` à la valeur `none`, faire disparaître de l'affichage toutes les images, tous les boutons y compris les boutons « twitter ».
- Transformer tous les items `li` de la liste à puce du bloc de navigation en des « inline-block » de largeur 15% et de marges intérieures haute et basse de 25px.
- Appliquez le style italique à tous les paragraphes des mentions légales

Pourquoi est-il préférable de privilégier le sélecteur enfant au sélecteur descendant ?

les sélecteurs d'adjacences

X ~ Y: Le sélecteur de voisins...

La sélection va se faire sur tous les Y qui suivent un élément X. De plus, il faut que les deux éléments soient au même niveau, disons plutôt qu'il ait le même père tous les deux.

```
p ~ span{
  background-color : green;
}
```

```
<div>
  <span> rien ! </span>
  <p class="paragraphe">
    <span> rien ! </span>
  </p>
  <span> green ! </span>
  <span> green ! </span>
</div>
<span> rien ! </span>
```

Les span qui sont après la balise p auront un fond vert. Notez qu'avec ce sélecteur, la règle ne s'appliquera ni à ceux qui sont avant l'élément X, donc la balise p dans notre exemple, ni à ceux qui ne sont pas du même niveau !

X + Y : Le sélecteur de voisins direct...

Même principe que pour le sélecteur précédent, mais ce sélecteur s'applique à l'élément directement proche de X ! La règle ne s'appliquera pas sur tous les autres Y qui suivent le premier.

```
p + span{
  background-color : yellow;
}
```

```
<div>
  <span> rien ! </span>
  <p class="paragraphe">
    <span> rien ! </span>
  </p>
  <span> yellow ! </span>
  <span> rien ! </span>
</div>
<span> rien ! </span>
```

Seul le span suivant directement la balise p aura un fond jaune !

- Appliquez une indentation de 10px au premier paragraphe qui suit un titre de niveau 4.
- Appliquez une marge intérieure gauche de 10px à tous les paragraphes précédés d'un titre de niveau 3.

Les selecteurs d'attribut

Chaque élément HTML peut avoir un ou plusieurs attributs. Ces attributs sont des caractéristiques supplémentaires (possédant souvent une plage de valeurs possibles) qui permettent de configurer l'élément ou de préciser son comportement. Par exemple :

```
<input type="text" required>
```

[name] est le **sélecteur d'attribut** qui permet de cibler toutes les balises de la page qui utilisent cet attribut.

```
input[type="text"] {
  /* Votre CSS personnalisé à insérer */
}
a[title] { background-color: red; }
```

Il est possible de cibler suivant tout ou partie de la valeur de l'attribut. Vous pourrez utiliser les symboles =, ~, |, ^, &=, *= qui ont chacun une spécificité. (que vous noterez)

- Mettre un fond jaune aux éléments qui ont l'attribut `data-percent`
- Colorez avec un fond de couleur rose les liens vers des ancres.
- Colorer en gris clair les liens pour téléchargement de fichier pdf.
- Colorer en rouge les liens externes.

les pseudo-classes et pseudo-éléments

Une **pseudo-classe** est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la déclaration.

La pseudo-classe `:hover`, par exemple, permettra d'appliquer une mise en forme spécifique lorsque l'utilisateur survole l'élément ciblé par le sélecteur (changer la couleur d'un bouton par exemple).

Il existe une trentaine de pseudo-classes :

```
:active, :checked, :disabled, :empty, :enabled, :first-child, :first-of-type, :focus, :hover, :in-range, :invalid, :lang(langue), :last-child, :last-of-type, :link, :not(sélecteur), :nth-child(n), :nth-last-child(n), :nth-last-of-type(n), :nth-of-type(n), :only-of-type, :only-child, :optional, :out-of-range, :read-only, :read-write, :required, :root, :target, :valid, :visited
```

Les **pseudo-éléments** CSS vont permettre de modifier l'apparence **d'une partie seulement d'un élément HTML**, ou encore **d'ajouter du contenu au début ou à la fin d'un certain élément HTML**. On peut citer notamment les pseudo-éléments `::first-letter`, `::first-line`, `::before`, `::after`.

<https://css.developpez.com/tutoriels/utiliser-nouveaux-selecteur-css-3/#LII-C>

<https://codepen.io/fosse17/pen/WqaZwy>

- Lors du survol, transformez l'item li de la liste à puce du bloc de navigation en l'élargissant à 20%, en changeant sa taille de police à 20pt et en mettant à zéro ses marges intérieures haute et basse.
- Faire que le premier paragraphe des mentions légales occupe 60% de la page en étant justifié et centré.
- Faire que les items pairs de la liste numérotée du bloc alternance est une couleur d'arrière-plan grise, les items impairs orange avec un texte centré.
- Appliquez à la première lettre du texte des éléments de classe pfblock-title une taille de 25pt et une couleur #ab4c98.

Cascade, héritage et poids des sélecteurs

Quelques définitions

En consultant la page :

https://developer.mozilla.org/fr/Apprendre/CSS/Introduction_à_CSS/La_cascade_et_l_héritage

Expliquez ce qu'est :

- La cascade en CSS
- L'héritage en CSS
- Les règles de priorité suivant le poids des sélecteurs

Les spécifications CSS proposent une classification des sélecteurs selon un barème de poids sous forme de notation à quatre chiffres :

Poids a : règle CSS déclarée à l'aide de l'attribut HTML style= au sein de l'élément

Poids b : sélecteur d'identifiant

Poids c : sélecteur de classe, d'attribut ([]) ou de pseudo-classes (:hover, :focus, :firstchild)

Poids d : sélecteurs d'élément (p, div, a, ...) ou de pseudo-élément (:first-letter, :before, :after, ...)

poids nul : parenté (>) ou adjacence (+) Par exemple :

```
#content nav li a.on:hover {  
    color: red;  
}
```

1 ID (#content), 2 classe+pseudo-classe (.on :hover), 3 éléments (nav li a) => 123

Il surpassera ce sélecteur:

```
nav.verte li.gras a.on:hover {  
    color: green;  
}
```

0 ID, 4 classe+pseudo-classe (.on :hover .verte .gras), 3 éléments (nav li a) => 43

mais pas celui-ci:

```
#content #nav a {  
    color: red;  
}
```

2 ID, 0 classe+pseudo-classe, 1 éléments (a) => 201

La déclaration !important

L'ajout d'un **!important**, entre la valeur et le ; permet de surpasser les règles de cascade et de spécificité (pour autant qu'il n'y ait pas d'autres valeurs comportant un !important auquel cas, on retrouve les règles de cascade et spécificités appliquées):

```
#content nav a {  
    color: red !important;  
}
```

Un jeu pour s'entrainer

<https://flukeout.github.io/>